Explicitly Context-Aware Publish/Subscribe with Context-Invariant Subscriptions

David Lizcano
School of Computer Science
Universidad Politécnica de Madrid
28660 - Boadilla del Monte, Madrid, Spain
dlizcano@fi.upm.es

Sebastian Ortega
School of Computer Science
Universidad Politécnica de Madrid
28660 - Boadilla del Monte, Madrid, Spain
sortega@fi.upm.es

Javier Soriano
School of Computer Science
Universidad Politécnica de Madrid
28660 - Boadilla del Monte, Madrid, Spain
jsoriano@fi.upm.es

Sergio Vavassori
School of Computer Science
Universidad Politécnica de Madrid
28660 - Boadilla del Monte, Madrid, Spain
svavassori@fi.upm.es

ABSTRACT

Although context could be exploited to improve the performance, elasticity and adaptation in most distributed systems that adopt the publish/subscribe (P/S) model of communication, only very few works have explored domains with highly dynamic context, whereas most adopted models are context agnostic. In this paper, we present the key design principles underlying a novel context-aware content-based P/S (CA-CBPS) model of communication, where the context is explicitly managed, focusing on the minimization of network overhead in domains with recurrent context changes thanks to contextual scoping. We highlight how we dealt with the main shortcomings of most of the current approaches. Our research is some of the first to study the problem of explicitly introducing context-awareness into the P/S model to capitalize on contextual information. The envisioned CA-CBPS middleware enables the cloud ecosystem of services to communicate very efficiently, in a decoupled, but contextually scoped fashion.

Categories and Subject Descriptors

C.2.4 [**Distributed Systems**]: Distributed applications; H.3.4 [**Systems and Software**]: Information networks

General Terms

Design, Performance

Keywords

Content-based publish/subscribe, context-awareness, contextual scoping, overlay network

1. INTRODUCTION

The cloud computing model [12] has attracted new efforts from the research community in the quest to obtain interaction models that facilitate the development of complex distributed applications and the integration of information in the cloud. Such cloud systems rely on complex interactions between diverse distributed components whose communication pattern should allow for Internet-wide scalability. Publish/Subscribe (P/S) has regained interest as a communication paradigm suitable for the cloud since it decouples in terms of time, space and synchronization [9] that help improving scalability by decoupling communication.

The P/S communication paradigm implies three entities: publishers, subscribers and intermediary brokers. Publishers asynchronously notify events and may advertise class of events. Subscribers manifest their interest in certain events. An intermediary broker or network of brokers, depending on the approach, take charge of message delivery. Content-based P/S [4] is an appealing variant in which subscribers interests are represented as filters on the publisher notifications. Such filters employ a expressiveness-limited language designed for matching event content as the notifications spread on the cloud. Complex interactions on the cloud can benefit the abstraction level of CBPS.

In a large fraction of the application domains in which the P/S model of communications has been adopted as a smart solution for spreading information among a sizable group of users, the publisher or subscriber context (e.g., location information, environmental data, operating data, user preferences, etc.), could if available be relevant information.

Cloud infrastructures are therefore evolving to support ubiquitous and context-aware computation and information integration. For this purpose, their communication technologies must take into account the situation and context in which the information is produced or consumed. Complex distributed applications can exploit context-awareness to improve their performance, elasticity and adaptation to surrounding environment changes. For example, services providing data could restrict the dissemination of their outcomes to certain consumers based on their context, and every entity could subscribe to information whose provider matches a contextual scope. That means the context has to be added to the P/S model and shared by publishers and subscribers.

Despite many promising proposals, a powerful, fully-fledged

context-aware P/S model is yet to be created. The research community has focused on the wireless sensor network (WSN) field [7, 8] but none of the most widely adopted CBPS middleware coherently offers context-awareness, as we will see in the analysis of current solutions. On the other hand, it is not unusual to see context merely encoded into published notifications and subscriptions, an approach, which, as we argue in this paper, leads to strong inefficiencies. Explicitly managing context radically improves the way routing is performed in CBPS middleware to achieve efficiency.

In this paper, we present the key design principles underlying a new context-aware P/S model of communication, where the context is explicitly managed, focusing on the minimization of network overhead in domains with recurrent context changes. In particular, in Section 2 we review the solutions and proposals of context-aware P/S models. Section 3 presents the shortcomings of current approaches. In Section 4, we describe a set of design principles conceived to deal with above issues. Finally, in Section 5, we conclude our proposal, and present future work in this area.

2. BACKGROUND

As we are exploring how to exploit context to improve P/S systems, this section will briefly introduce key aspects about context management and, especially, context-aware CBPS systems.

Context-aware models are usually defined by how they react to context changes: *passively* or *actively* [5]. When passive models learn about a context change, they prompt the user before applying any change, whereas active models manage changes without user interaction, enabling automatic contextual reconfiguration. Another way of classifying these models is by how a context-aware application realizes the context has changed: this it can do by *sensing* the environment or by *being notified*. Notification has the advantage of reducing communication overhead but at the risk of missing changes.

Syntactically, context can be transmitted using different message structures and syntaxes, the most widespread [2] in increasing order of complexity being: key-value pairs, markup scheme models, graphical models, object oriented models, logic based models, and ontology based models. The flexibility/meaning trade-off differs from one representation to another. Key-value pairs representation is the most suitable option for integration with CBPS systems since it is the canonical representation of such systems and, at the same time, has a rich internal representation for context modeling.

To deal with context [14] in CBPS systems, context information has to be processed from the viewpoints of both the subscriber and the publisher. Subscriber contextual information provides for filtering according to user location, device, current situation, preferences, etc., in order to get relevant, useful and appropriate information within the context. Publisher contextual information provides for message adaptation according to location or other context factors to get relevant, useful and appropriate information. Routing algorithms should bridge the gap, while leveraging both context scopes to reduce network overhead.

Context-awareness is a recent research line in the CBPS field, where most proposals are simply context agnostic. Within the context-aware approaches, most of the work has focused on implicitly context-aware strategies [11], which piggyback context on notifications or communicate context at subscription time.

Most recent approaches have explored explicit representation of context through an extended API to set the context that the broker network uses for notification routing and topology shaping [8]. However, these approaches have context-coupled subscriptions since both context and content filters are entangled making difficult them to manage independently.

Finally, systems can vary in terms of the aspects considered as context information. Most mature projects available largely equate context with location management and are known as location based services (LBS). L-ToPSS [3] is a LBS system based on a P/S middleware that adds an extra location processing module to a typical event broker to manage the possible spatial events and subscriptions. The system aims at supporting window queries and N-nearest queries. Based on L-ToPSS, [13] proposes efficient algorithms for location constraint evaluation. CAMEL [6] is a push-based middleware construct based on a database. Similar to L-ToPSS, the system is designed to support window queries and N-nearest queries. Most of the research investigates how the constraints or predicates in subscriptions can be evaluated more efficiently, but little progress has been made with respect to enhancing the expressiveness of spatial subscriptions [10].

3. SHORTCOMINGS OF IMPLICIT CONTEXT MANAGEMENT

Taking into account the roles played, notifications and subscriptions, the problem is necessarily strongly affected by the context of the entities, in particular, their location. A naïve application of CBPS to such a scenario would consists on merging content and context in both messages and filters, it becomes context-aware in a implicit manner. An in-depth analysis unveils several shortcomings that are discussed in this section.

- **Matching inversion.** Classic CBPS systems models messages as key-value sets, $m \in \{\mathbb{K} \times \mathbb{V}\}$, which can be matched by filters issued by the subscribers, $f \in F \subseteq \{\mathbb{K} \times \mathbb{V}\}$. If m is in the scope of a filter f, we can state that f covers m or $f \succeq m$. The purpose of the message is to encode data, whereas the filter represents the interests in this information.

  Applying CBPS to context-aware scenarios violates this separation of concerns. On the one hand, the scenario includes several subscription filters in which subscriber contextual information is embedded On the other hand, published notifications include restrictions that the subscribers must enforce, abusing the concept of notification, which represents an event and nothing else. This implies a change in data models and CBPS inference, reversing the classical matching process.

  Only a few CBPS systems can afford this out-of-the-box change, those who implement filtering using Turing-

complete languages. Anyway, such systems are difficult to optimize and exploit in the first place.

- **Message flooding in changing environments.** Published messages bundle the notification content with contextual information ($\mathbb{K} = \mathbb{K}_{content} \cap \mathbb{K}_{ctx}$), for example, the events detected by cameras include the camera location. This approach has a comparatively small overhead with respect to a context-agnostic scenario since the rate of change of the context is small compared with the rate of publication.

  If that premise is not met as in Figure 1, constant context updates will create a massive overhead because existing subscription filters (①) are constantly invalidated (②) and then updated (③). In terms of messages, this implies a waterfall of unsubscriptions and new updated subscriptions flooding the whole network as illustrated. In our scenario, driver subscriptions pose this problem since their position changes constantly, and they are encoded as part of the filter.

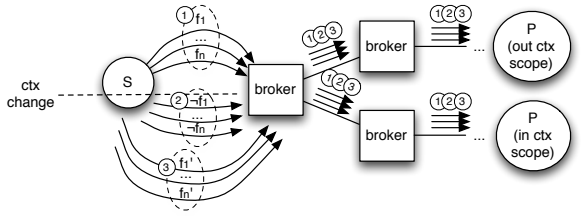

Figure 1: Exchange of messages with implicit context management.

- **Unrealized potential performance.** If properly exploited, contextual information, can improve efficiency by minimizing networking overhead in two senses: on the one hand, messages could be efficiently routed; on the other hand, the overlay network topology could be optimized. For instance, clustering drivers interested in *nearby traffic alerts* (same interest) is of no use without consider the context that permeates their interest. Similarly, clustering drivers at nearby locations (same context) is plagued by the same problem, since it ignores their interests.

  Using implicit context however, the broker will have to handle more complexity in the form of bigger routing tables and longer routing computation.

  Finally, any advantage from context exploitation requires explicit context management. Implicit context-aware solutions require the broker to parse both messages and filters, which is an additional overhead.

- **Separation of concerns within entities.** Context management and subscription or publication are very different concerns that must be handled by different components at the architectural level. Implicit context-awareness favors coupling and forces rigid designs in which the P/S component is also responsible for detecting, managing and communicating context changes.

  For instance, drivers can use a mobile application in which location context is managed by a GPS component decoupled from the subscriber component, in charge of handling incoming notifications.

- **Undermined publisher/subscriber decoupling.** The main property of CBPS systems is decoupling in terms of space, time and synchronization between publisher and subscriber. The only thing the subscriber should know is how the events of interest are represented.

  Subscribers and publishers are obliged to share a unified and homogeneous representation of context, as context is embedded in notifications for implicitly context-aware systems. In the example scenario, location is represented uniformly as a coordinate point for all the entities. A more flexible approach, such as ontology-based mediation, is beyond the scope of the scheme.

4. PROPOSED DESIGN PRINCIPLES

There is a clear motivation to overcome the weaknesses of implicitly context-aware management, which deteriorate as the application scenario context becomes more changeable.

Our approach consists of decoupling context and content throughout the CBPS with the aim of decoupling publishers from subscribers in space, time and synchronization within a *contextual scope*.

The design principles that guide our vision of a Context-Aware Content-Based Publish/Subscribe (CA-CBPS) are as follows:

1. *Explicit separation of context and content.* The scope of both messages and their filters must be restricted and deal exclusively with content. Therefore, $m \in \{\mathbb{K}_{content} \times \mathbb{V}\}$.

2. *Generalized P/S model for contextual scoping.* The reversal of the matching procedure demonstrates an impedance mismatching between content-based filtering and any other type of filtering that does not fit the content-based model. Symmetric context-scoping filters will be attached to both subscriptions and publications, restricting publisher contexts and subscriber contexts, respectively. Such filters will be attached ($||$) to publications, $m||f_{cs},\ f_{cs} \in F_{ctx}$, and to subscriptions $f||f_{cp},\ f_{cp} \in F_{ctx}$ given $F_{ctx} \subseteq \{\mathbb{K}_{context} \times \mathbb{V}\}$. Note how publisher and subscriber can restrict their actions symmetrically by each other's contextual scope. This scoping mechanism can be seen as a role-based access control by the publisher [1] and as new filtering dimension by the subscriber.

3. *Context-invariant subscriptions and advertisements.* Subscriptions will be context-invariant to avoid context changes flooding. Context variable references will be used instead of embedding context values as part of the filters. In fact, a subscription f_s defined as $f||f_{cp}$ can be regarded as a function on the context in which the filter f_{cp} is the context-aware part.

$$f_s : \mathbb{C} \rightarrow F$$
$$f_s(ctx) \subseteq \{\mathbb{K} \times \mathbb{V}\}$$

Such functions become regular filters when applied to a given context $ctx \in \mathbb{C}$ within the brokers. Figure 2 illustrates how flooding is avoided as subscriptions (①) are updated by propagating context updates (②).

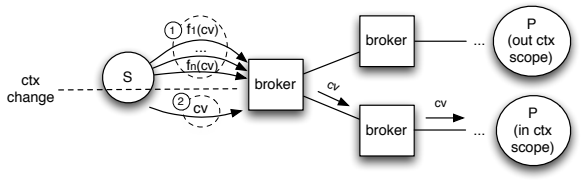

Figure 2: Exchange of messages with our proposed explicit context management.

4. *Context is distributed in terms of opaque, independent variables.* Context is distributed in the form of individual variables to support an optimized update policy. These variables have unique identifiers and no semantics so they can be cached and processed as plain values. Their meaning is embedded in the function filters.

   Intermediate brokers can exploit this design principle to analyze, abstract, aggregate or perform other arbitrary transforms to minimize network overhead or other performance metrics.

5. *Ultimate context-variable management is the responsibility of the border brokers.* Opaque variables refer, possibly, to a managed context maintained by the border broker to which the entity is connected. This broker *owns* the entity context and is responsible for forwarding updates.

5. CONCLUSION AND FUTURE WORK

As we have shown our model adds a new dimension to CBPS flexibility: context scoping based on context-invariant filters that are extremely valuable for domains with highly dynamic context.

Cloud computing will capitalize on this proposal in several areas. On the one hand, services being coordinated through a P/S communication paradigm will be able to be adapted by leveraging their context information. On the other hand, the decoupling and flexibility of the CA-CBPS model will bring cloud computing closer to realizing ubiquitous computing. Finally, the proposed asynchronous context-aware communication transforms the cloud infraestructures into the ideal ecosystem of autonomous services.

Our approach does not yet consider aspects like privacy and access control, which are orthogonal to the context-awareness of the CBPS and are different research lines that can be validated together in the future.

Future work will concentrate on exploring the effect of subscriber to publisher ratio on the optimal network topology and routing strategies; subscriber (or publisher) handover to minimize network restructuring; and how to exploit the opaque variable mechanism for optimization.

Finally, note that the presented work is being developed and applied as part of the 4CaaSt project[1], a platform as a service, where P/S is offered as both a value-added service to hosted applications and a key internal platform asset.

6. ACKNOWLEDGMENTS

This work is partially supported by the 7th FP EU project 4CaaSt (Building the PaaS Cloud of the Future) with grant agreement no. 258862.

7. REFERENCES

[1] J. Bacon, D. M. Eyers, J. Singh, and P. R. Pietzuch. Access control in publish/subscribe systems. In *Proc. of the 2nd int. conference on Distributed event-based systems*, DEBS, pages 23–34, NY, USA, 2008.

[2] M. Baldauf and S. Dustdar. A survey on context-aware systems. *Int. Journal of Ad-hoc and Ubiquitous Computing*, page 2004, 2004.

[3] I. Burcea and H.-A. Jacobsen. L-topss - push-oriented location-based services. In *TES*, pages 131–142, 2003.

[4] A. Carzaniga, D. Rosenblum, and A. Wolf. Design and evaluation of a wide-area event notification service. *ACM TonCS*, 19(3):332–383, 2001.

[5] G. Chen and D. Kotz. A survey of context-aware mobile computing research. Technical report, Hanover, NH, USA, 2000.

[6] X. Chen, Y. Chen, and F. Rao. An efficient spatial p/s system for intelligent location-based services. In *Proc. of the 2nd Int. workshop on Distributed event-based systems*, DEBS, pages 1–6, NY, USA, 2003.

[7] G. Cugola and J. de Cote. On introducing location awareness in publish-subscribe middleware. In *Distributed Computing Systems Workshops, 2005. 25th IEEE Int. Conference on*, pages 377 – 382, june 2005.

[8] G. Cugola, A. Margara, and M. Migliavacca. Context-aware publish-subscribe: Model, implementation, and evaluation. In *Computers and Communications, 2009. ISCC 2009. IEEE Symposium on*, pages 875 –881, july 2009.

[9] P. T. Eugster, P. A. Felber, R. Guerraoui, and A.-M. Kermarrec. The Many Faces of Publish/Subscribe. *ACM Comput. Surv.*, 35(2):114–131, 2003.

[10] K.-K. Fu. Mobile spatial subscriptions for location-aware services. Master's thesis, Master of Mathematics in Computer Science Thesis University of Waterloo, Waterloo, Ontario, Canada, 2010.

[11] G. M. Kapitsaki, G. N. Prezerakos, N. D. Tselikas, and I. S. Venieris. Context-aware service engineering: A survey. *J. Syst. Softw.*, 82:1285–1297, August 2009.

[12] L. Vaquero, L. Rodero-Merino, J. Caceres, and M. Lindner. A break in the clouds: towards a cloud definition. *ACM SIGCOMM Computer Communication Review*, 39(1):50–55, 2008.

[13] Z. Xu and H.-A. Jacobsen. Efficient constraint processing for highly personalized location based services. In *Proc. of the Thirtieth Int. conference on Very large data bases - Volume 30*, VLDB, pages 1285–1288. VLDB Endowment, 2004.

[14] J. yi Hong, E. ho Suh, and S. jin Kim. Context-aware systems: A literature review and classification, 2009.

[1] http://4caast.morfeo-project.org/